

Day 1: Cyber Security Internship

NAME:- Digvijay Netke

Task 1 : scan your local network with Open port

Objective: Learn to discover open ports on devices in your local network to understand network exposure.

Tools: Nmap (free), Wireshark (optional).

1. Install Nmap from official website

=> already installed

└─\$ nmap --version

Nmap version 7.95 (<https://nmap.org>)

Platform: x86_64-pc-linux-gnu

Compiled with: liblua-5.4.7 openssl-3.5.3 libssh2-1.11.1 libz-1.3.1 libpcap-1.10.5

nmap-libdnet-1.12 ipv6

Compiled without:

Available nsock engines: epoll poll select

2. Find your local IP range (e.g., 192.168.1.0/24).

=>

└─\$ ip a s

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid_lft forever preferred_lft forever

inet6 ::1/128 scope host noprefixroute

valid_lft forever preferred_lft forever

2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000

link/ether 58:11:22:ea:3b:d8 brd ff:ff:ff:ff:ff:ff

3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000

link/ether 2c:3b:70:6c:b6:65 brd ff:ff:ff:ff:ff:ff

inet 10.235.113.94/24 brd 10.235.113.255 scope global dynamic noprefixroute wlan0

valid_lft 2816sec preferred_lft 2816sec

inet6 2409:4081:1016:8537:aa84:41f1:f39f:794c/64 scope global dynamic noprefixroute

valid_lft 6877sec preferred_lft 6877sec

inet6 fe80::4b28:c98b:7466:fb74/64 scope link noprefixroute

valid_lft forever preferred_lft forever

3. Run: nmap -sS 192.168.1.0/24 to perform TCP SYN scan.

=>

└─\$ nmap -sS 10.235.113.0/24

Starting Nmap 7.95 (<https://nmap.org>) at 2025-10-20 22:08 IST

Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan

ARP Ping Scan Timing: About 27.45% done; ETC: 22:08 (0:00:08 remaining)

Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
 ARP Ping Scan Timing: About 62.75% done; ETC: 22:08 (0:00:04 remaining)
 Nmap scan report for 10.235.113.74
 Host is up (0.0046s latency).
 Not shown: 999 closed tcp ports (reset)
 PORT STATE SERVICE
 53/tcp open domain
 MAC Address: 2A:CB:ED:B6:46:14 (Unknown)

Nmap scan report for 10.235.113.94
 Host is up (0.000015s latency).
 All 1000 scanned ports on 10.235.113.94 are in ignored states.
 Not shown: 1000 closed tcp ports (reset)

4. Note down IP addresses and open ports found.

=>
 IP – 10.235.113.74
 port – 53

5. Optionally analyze packet capture with Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
1517	8.878095641	2409:4081:1016:8537...	2404:6800:4009:80d:...	QUIC	717	Protected Payload (KP0), DCID=fb4
1518	8.878287151	2409:4081:1016:8537...	2404:6800:4009:80d:...	QUIC	1292	Protected Payload (KP0), DCID=fb4
1519	8.878338555	2409:4081:1016:8537...	2404:6800:4009:80d:...	QUIC	213	Protected Payload (KP0), DCID=fb4
1520	8.878536281	2409:4081:1016:8537...	2404:6800:4009:80d:...	QUIC	765	Protected Payload (KP0), DCID=fb4
1521	8.878751677	2409:4081:1016:8537...	2404:6800:4009:80d:...	QUIC	1234	Protected Payload (KP0), DCID=fb4
1522	8.910551759	10.235.113.94	239.255.255.250	SSDP	214	M-SEARCH * HTTP/1.1
1523	8.923801146	10.235.113.94	10.235.113.74	DNS	79	Standard query 0x659a HTTPS accou
1524	8.923913803	10.235.113.94	10.235.113.74	DNS	79	Standard query 0x192e AAAA accoun
1525	8.923982529	10.235.113.94	10.235.113.74	DNS	79	Standard query 0x3c17 A accounts.
1526	8.924966968	2409:4081:1016:8537...	2404:6800:4009:80d:...	QUIC	280	Protected Payload (KP0), DCID=fb4
1527	8.954454612	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	93	Protected Payload (KP0)
1528	8.954456009	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	93	Protected Payload (KP0)
1529	8.954729026	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	93	Protected Payload (KP0)
1530	8.958703521	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	1287	Protected Payload (KP0)
1531	8.958704918	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	1292	Protected Payload (KP0)
1532	8.959020330	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	178	Protected Payload (KP0)
1533	8.959021308	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	1287	Protected Payload (KP0)
1534	8.959048756	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	1292	Protected Payload (KP0)
1535	8.959049734	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	1292	Protected Payload (KP0)
1536	8.959050642	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	1131	Protected Payload (KP0)
1537	8.967529741	2409:4081:1016:8537...	2404:6800:4009:80d:...	QUIC	101	Protected Payload (KP0), DCID=fb4
1538	8.975419435	10.235.113.74	10.235.113.94	DNS	129	Standard query response 0x659a HT
1539	8.985427820	10.235.113.74	10.235.113.94	DNS	107	Standard query response 0x192e AA
1540	8.985971270	10.235.113.74	10.235.113.94	DNS	95	Standard query response 0x3c17 A
1541	8.987302969	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	1292	Initial, DCID=66f287636ff3fb15, P
1542	8.987346202	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	1292	Initial, DCID=66f287636ff3fb15, P
1543	8.987393766	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	1292	Initial, DCID=66f287636ff3fb15, P
1544	8.990430616	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	143	0-RTT, DCID=66f287636ff3fb15
1545	8.990724377	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	1288	0-RTT, DCID=66f287636ff3fb15
1546	8.990747495	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	1292	0-RTT, DCID=66f287636ff3fb15
1547	8.990765724	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	1292	0-RTT, DCID=66f287636ff3fb15
1548	8.990813217	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	964	0-RTT, DCID=66f287636ff3fb15
1549	8.990833821	2409:4081:1016:8537...	2404:6800:4003:c1a:...	QUIC	153	0-RTT, DCID=66f287636ff3fb15
1550	9.013155696	54.71.154.29	10.235.113.94	TCP	74	443 → 54540 [SYN, ACK] Seq=0 Ack=
1551	9.013203748	10.235.113.94	54.71.154.29	TCP	54	54540 → 443 [RST] Seq=1 Win=0 Len
1552	9.014057022	2404:6800:4009:80d:...	2409:4081:1016:8537...	QUIC	95	Protected Payload (KP0)
1553	9.019134271	64:ff9b::2ce2:9e37	2409:4081:1016:8537...	TCP	86	443 → 52452 [ACK] Seq=1 Ack=1359
1554	9.019135179	64:ff9b::2ce2:9e37	2409:4081:1016:8537...	TCP	86	443 → 52452 [ACK] Seq=1 Ack=1790
1555	9.022101418	64:ff9b::2ce2:9e37	2409:4081:1016:8537...	TLSv1.3	2762	Server Hello, Change Cipher Spec

6. Research common services running on those ports

```

- Frame 4858: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlan0, id 0
  Section number: 1
  ▸ Interface id: 0 (wlan0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Oct 20, 2025 22:11:47.580057512 IST
    UTC Arrival Time: Oct 20, 2025 16:41:47.580057512 UTC
    Epoch Arrival Time: 1760978507.580057512
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.974445490 seconds]
    [Time delta from previous displayed frame: 0.974445490 seconds]
    [Time since reference or first frame: 60.118753114 seconds]
    Frame Number: 4858
    Frame Length: 42 bytes (336 bits)
    Capture Length: 42 bytes (336 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
  ▸ Ethernet II, Src: 2a:cb:ed:b6:46:14 (2a:cb:ed:b6:46:14), Dst: AzureWaveTec_6c:b6:65 (2c:3b:70:6c:b6:65)
    ▸ Destination: AzureWaveTec_6c:b6:65 (2c:3b:70:6c:b6:65)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
    ▸ Source: 2a:cb:ed:b6:46:14 (2a:cb:ed:b6:46:14)
      Type: ARP (0x0806)
      [Stream index: 0]
  ▸ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 2a:cb:ed:b6:46:14 (2a:cb:ed:b6:46:14)
    Sender IP address: 10.235.113.74
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.235.113.94

```

7. Identify potential security risks from open ports.


```

1502 8.787758188 64:ff9b::2ce2:9e37 2409:4081:1016:8537... TCP 1424 443 → 52448 [PSH, ACK] Seq=133
1503 8.787783472 2409:4081:1016:8537... 64:ff9b::2ce2:9e37 TCP 86 52448 → 443 [ACK] Seq=1822 Ack
1505 8.788250722 2409:4081:1016:8537... 64:ff9b::2ce2:9e37 TCP 86 52448 → 443 [ACK] Seq=1822 Ack
1550 9.013155696 54.71.154.29 10.235.113.94 TCP 74 443 → 54540 [SYN, ACK] Seq=0 A
1551 9.013203748 10.235.113.94 54.71.154.29 TCP 54 54540 → 443 [RST] Seq=1 Win=0
1553 9.019134271 64:ff9b::2ce2:9e37 2409:4081:1016:8537... TCP 86 443 → 52452 [ACK] Seq=1 Ack=13
1554 9.019135179 64:ff9b::2ce2:9e37 2409:4081:1016:8537... TCP 86 443 → 52452 [ACK] Seq=1 Ack=17
1556 9.022137596 2409:4081:1016:8537... 64:ff9b::2ce2:9e37 TCP 86 52452 → 443 [ACK] Seq=1790 Ack
1558 9.022325684 2409:4081:1016:8537... 64:ff9b::2ce2:9e37 TCP 86 52452 → 443 [ACK] Seq=1790 Ack
1657 9.112047678 64:ff9b::2ce2:9e37 2409:4081:1016:8537... TCP 86 443 → 52448 [ACK] Seq=3142 Ack
1660 9.112406742 2409:4081:1016:8537... 64:ff9b::2ce2:9e37 TCP 86 52448 → 443 [ACK] Seq=2675 Ack
1684 9.137169077 2409:4081:1016:8537... 64:ff9b::2ce2:9e37 TCP 86 52448 → 443 [ACK] Seq=2706 Ack
1812 9.324118621 2409:4081:1016:8537... 64:ff9b::2ce2:9e37 TCP 86 52452 → 443 [ACK] Seq=1870 Ack
1829 9.360173338 2409:4081:1016:8537... 2404:6800:4009:823:... TCP 94 36426 → 443 [SYN] Seq=0 Win=64
1836 9.422121919 2404:6800:4009:823:... 2409:4081:1016:8537... TCP 94 443 → 36426 [SYN, ACK] Seq=0 A
1837 9.422169483 2409:4081:1016:8537... 2404:6800:4009:823:... TCP 86 36426 → 443 [ACK] Seq=1 Ack=1
1838 9.422797652 2409:4081:1016:8537... 2404:6800:4009:823:... TCP 1444 36426 → 443 [ACK] Seq=1 Ack=1
1857 9.448468996 64:ff9b::2ce2:9e37 2409:4081:1016:8537... TCP 86 443 → 52448 [ACK] Seq=4100 Ack
1869 9.453526618 2409:4081:1016:8537... 2404:6800:4009:806:... TCP 94 41658 → 443 [SYN] Seq=0 Win=64
Destination Address: 10.235.113.94
[Stream index: 0]
> User Datagram Protocol, Src Port: 10943, Dst Port: 53
> Domain Name System (query)
Transaction ID: 0xf4aa
> Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
> Queries
[Response In: 26]

```

8. Save scan results as a text or HTML file

=>

https://drive.google.com/file/d/1hGlofTKXt8_-1ebV3WmW5kmQQzuofvf3/view?usp=sharing

```

total 110928
drwxrwxr-x 2 ryzen ryzen 4096 Oct 20 22:18 .
drwxr-xr-x 8 root root 4096 Oct 10 11:33 ..
-rw-rw-r-- 1 ryzen ryzen 1418820 Oct 9 12:55 NIPS-2012-imagenet-classification-with-deep-convolutional-neural-networks-Paper.pdf
-rw-rw-r-- 1 root root 112157715 Oct 20 22:18 task1_20oct2025.pdml

```

[\(source\)](#)

interview

INTERVIEW QUESTIONS

What is an open port?

An open port is a network port on a host that is accepting incoming connections or datagrams. Ports are logical endpoints identified by a number (0–65535) that let services (like a web server, SSH, or a game server) communicate over TCP or UDP. If a port is open, a service is listening on that port and can respond to network requests.

1. How does Nmap perform a TCP SYN scan?

In a TCP SYN scan (often called a “half-open” scan), Nmap sends a TCP packet with only the SYN flag set to the target port. The target’s response reveals the port state:

- **SYN/ACK** → the port is **open** (Nmap then sends an RST to avoid completing the handshake).
- **RST** → the port is **closed**.
- **No response or ICMP unreachable** → the port is **filtered** (likely blocked by a firewall).

This method is fast and stealthier than completing the full TCP handshake because it avoids establishing a full connection.

2. **What risks are associated with open ports?**

- **Attack surface:** Each open port exposes a service that might have vulnerabilities (bugs, weak auth, misconfiguration).
- **Unauthorized access:** If services aren't properly secured, attackers can exploit them to gain access.
- **Information leakage:** Open ports can reveal software versions or service banners that help attackers craft targeted attacks.
- **Pivoting:** Compromised services can let attackers move laterally inside a network.
- **Denial-of-service:** Public-facing services can be overwhelmed via the open port.

3. **Explain the difference between TCP and UDP scanning.**

- **TCP scanning** (e.g., SYN, connect) targets connection-oriented TCP ports. Responses are explicit: SYN/ACK, RST, etc., making results usually reliable. TCP scans often determine whether a service accepts and tries to establish connections.
- **UDP scanning** targets connectionless UDP ports. UDP has no handshake; many services don't reply when closed. Nmap typically sends an empty UDP packet and interprets responses (ICMP port unreachable = closed; no response = open|filtered). UDP scanning is slower and less definitive because of rate-limiting and silent drops.

4. **How can open ports be secured?**

- **Close unnecessary ports:** Disable or uninstall unused services.
- **Use firewalls:** Restrict which IPs/subnets can reach ports.
- **Strong authentication & least privilege:** Use strong credentials, keys, and minimal permissions.
- **Keep services patched:** Regularly update software to fix vulnerabilities.
- **Service hardening:** Disable verbose banners, enforce encryption (TLS), and apply rate-limiting.
- **Network segmentation:** Keep critical systems separated from public-facing networks.
- **Monitoring & logging:** Detect suspicious access patterns and respond quickly.

5. **What is a firewall's role regarding ports?**

A firewall controls traffic to and from a host or network by allowing, denying, or restricting connections to specific ports and protocols. It enforces access rules (e.g., allow SSH only from admin IPs), provides a first line of defense by filtering unwanted traffic, and can log attempts for auditing and detection.

6. **What is a port scan and why do attackers perform it?**

A port scan is the process of probing a host or network to discover which ports are open and which services run on them. Attackers use port scans to:

- **Identify attackable services** and their versions.
- **Map an environment** for lateral movement.
- **Prioritize targets** for exploitation.

Security teams also use port scanning defensively to find exposed services and close or secure them.

7. **How does Wireshark complement port scanning?**

Wireshark captures and analyzes live network traffic at a packet level. It complements port scanning by:

- **Verifying traffic:** Showing the actual packets exchanged during a scan or an exploit attempt.
- **Troubleshooting responses:** Helping interpret ambiguous scan results (e.g., why a probe was dropped).
- **Detecting anomalies:** Revealing unexpected services, malformed packets, or suspicious flows that a port scan alone might miss.
- **Forensics:** Providing detailed logs of attacker activity or misconfigurations for post-incident analysis.